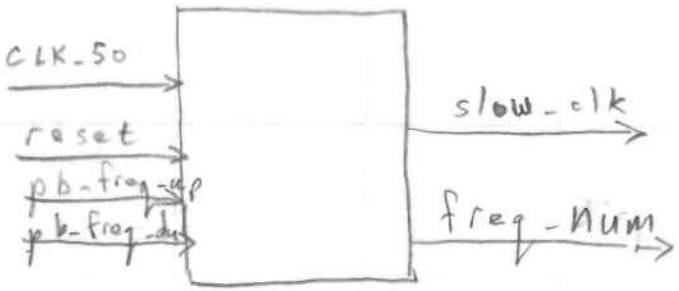
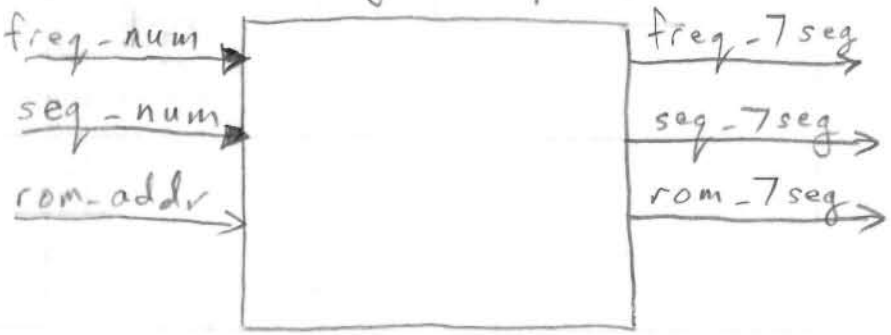
KROS

7/11/2016

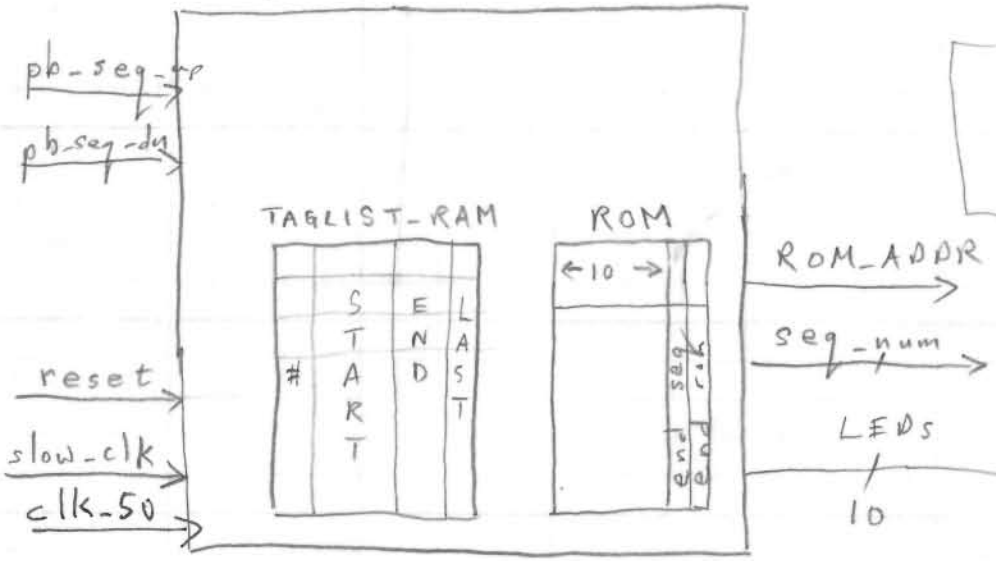
throttle

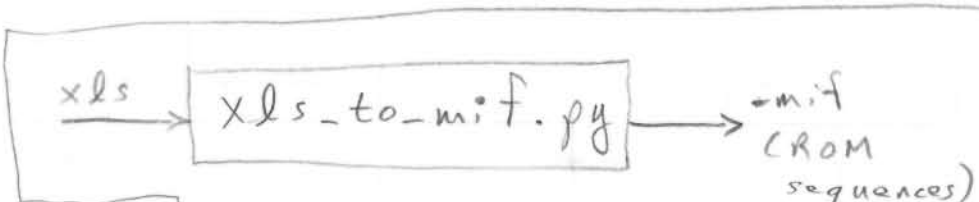


seg_disp



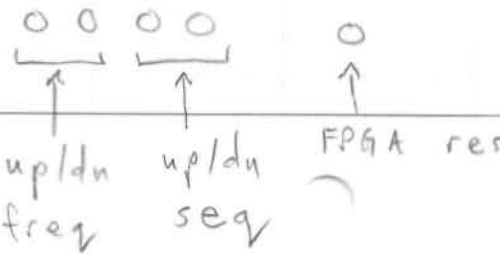
sequencer





Theory of operation:

The ROM stores variable # of sequences, each 10 bits. Each 10 bit word has end_seq & end_rom tag. PBs adv/decrease freq and sequences.



7/11/2016

1) Team decides various functions:

1) Top level integrator
2) Sequencer coder (1 or 2 people)
   *toughest module pb up/dn
3) seg_disp coder
   hex → bcd → 7 segment
4) throttle coder - 50 Mhz in to variable frequency out based on pb up/dn
5) xls to mif python coder (also write sequences)
6) Overall coordinator - general Q&A and submodule testbenches
7) Bonus function: Add audio

Hints:
- Sequencer most complex module.. might consider using state machine designer.
- The ROM and taglist RAM need to be dual ported. The clocks are 50 Mhz and the divided clock.
- Feel free to grab IP, shareware or other ways to speed up implementation
- Push buttons should be debounced and edge detected
- It's easier to find bugs with the simulator
- Agree on signal I/O & naming conventions early